Screencast: 14-systemd.webm or 14-systemd.mp4

Alternative init systems

There are other multiple init systems used by the various Linux distributions. Some distributions, particularly those styled after BSD, may use a single config file that has to be updated whenever services are added/removed.

A former Canonical (the company that sponsors the Ubuntu Linux distribution) employee named **Scott James Remnant** created **upstart**. Ubuntu historically had a SysV-based init system but switched to upstart several releases ago.

Fedora historically used a SysV-based init system. After upstart came out, they switched to it... although at the time upstart was not very complete and was primarily configured in SysV-compatibility mode.

5+ years ago, Red Hat employee and Fedora developer **Lennart Poettering** did a survey of the various init systems available for various flavors of Unix/Linux. He compared features and performance and then created a new init system named **systemd**. systemd is designed to be a modern init system specifically for Linux... and as such it takes advantage of some Linux-only features... which makes it less portable. Some advanced things that systemd does is parallel execution, socket and dbus service activation, automatic cgroup resource management / scheduling, and multi-seat support. It has a system boot profiler with graphing and by disabling unneeded (and / or slow) services, you can greatly speed up your boot times. systemd has had "feature creep" set in and it has expanded beyond traditional init features. Another one of its goals is to do common system configuration (hostname, file system mounting, etc) and do it in a distribution agnostic way. systemd also has man security enhancement features.

Many distributions have switched to systemd including all of the top, mainstream distributions like RHEL, Fedora, Debian, Ubuntu, Mint, Arch, etc. gentoo defaults to a different init system named OpenRC. A group of disgruntled-over-systemd users forked Debian and created Devuan.

systemd

cgroups - As I talked about near the end of the Process and Resource Management lecture, systemd is an enabler for cgroups and starts everything in a cgroup. As a result the resource usage of CPU, RAM, and DISK are dynamically tunable (network coming in the future). cgroups makes it easier and faster to reliably stop services.

journald - systemd decided to take on the logging facilities and as a result of the logging system being integrated into the init system and being able to have better access to the kernel, it can log all kernel messages including full startup and shutdown... a feat that was previously impossible. It also uses a signle binary log file with database like functionality. You can still run a traditional syslog program in parallel if desired to get the old-school, standard single log file per service text logs. more abut logging in a separate lecture.

unit and target files - The traditional / original Unix / Linux init system SysVinit used shell scripts to control all services and there are many drawbacks. systemd abandoned shell scripts and has much smaller, declarative configuration files named **unit files**. SysVinit init had runlevels. systemd abandoned runlevels and uses **target files** instead. Available targets are emergency, rescue, multi-user (text-console only) and graphical (like multi-user but with a graphical login added).

core os concept - systemd decided to go beyond the strict boundaries of the traditional init system to incorporate more features that made sense for a system trying to make services and resources manageable. The expansion of features has been dubbed, "Core OS" (not to be confused with the CoreOS Linux distribution) in that it tries to make common tasks the same across all distributions that use systemd. There used to be several different programs used by distros for such things as logging in, managing user sessions, watchdog, cron, logging, setting the hostname etc. This the most contentious aspect of systemd as some users think it has gone to far.

systemd programs / commands

The main control program for systemd is *systemctl*.

systemctl --full --all (lists everything)
systemctl enable {servicename}
systemctl disable {servicename}
systemctl status {servicename}
systemctl start {servicename}
systemctl stop {servicename}
systemctl restart {servicename}
systemctl mask {servicename (sysmlinks to /dev/null)
systemctl get-default (shows default target)
systemctl set-default {targetname} (changes the default target)
sysmtectl isolate {targetname} (change target)
sytemctl edit (add drop-in files for service customization)